

METHOD AND SYSTEM FOR AUTOMATIC CORRECTION OF MOTION ARTIFACTS

5

FIELD OF THE INVENTION

The present invention relates to methods and systems for improving interlaced video and images taken from endoscopic or other surgery, sports still frames, or other video. In particular, the methods and systems of the present invention relate to improving interlaced video and images in which the image recording device, subject, or both are moving independently. The present invention more particularly relates to methods and systems for automatically correcting artifacts caused by camera motion and artifacts caused by subject motion.

BACKGROUND OF THE INVENTION

Television and video technology has long relied on trickery to fool the human eye into believing that the television or video signal accurately reflects the events occurring. For instance, conventional video technology such as the NTSC (National Television Standards Committee) standard uses "interlaced video," in which a single image or frame comprises two fields that are taken 1/60 second apart. Each frame contains 525 scan lines divided into two fields. The first, or odd, field comprises the odd numbered lines (e.g., 1, 3, ... 525) while the second, even field forms the even numbered lines (e.g., 2, 4, ... 524). These two fields, during display of the frame they form, are interlaced so that the odd numbered scan lines are inserted between the even numbered scan lines, much like the interlacing that would occur when one interlaces

the fingers of one hand with the fingers of another. Since each frame comprises two interlaced fields (even and odd) each captured 1/60 second apart, the frames themselves are captured at a rate of 30 frames per second.

In most video applications, the very short delay (1/60 second) between capture
5 of fields means that even if the subject moves during filming, any artifacts of motion will be virtually undetectable. (An example of motion artifacts can be seen in old silent movies, which operated at fairly low frame speeds.) The high rate at which the recorder captures frames and the slight time separation between the fields within each frame results in minimal blurring in the video. Even when the subject being video-recorded is
10 in motion, for example, a moving car or a runner, such motion artifact will be substantially undetectable to the human eye in the captured frame. Additionally, in many applications, humans tend to ignore even detectable motion artifacts in video because a particular motion artifact (for example, blurring because of subject movement) is often quickly replaced with another frame or series of frames missing that
15 artifact. Thus, in some applications, the speed of frame capture and the continual refreshment of interlaced fields is sufficient to avoid noticeable blurring of video images.

However, in many applications, the rate at which fields are scanned and frames are captured is not sufficiently high to prevent motion artifacts from causing image distortion problems in the displayed video. Applications in which image distortion
20 problems may be caused by motion artifacts include, for example, video and images taken from endoscopic or other surgery, sports still frames, or other video in which the camera and subject are moving independently. U.S. Patent No. 5,191,413 to Edgar

gives a practical example of this. Edgar states that "if a subject were to move a hand during the capturing process, sequential fields will be generated which will capture the hand in two distinctly different positions. However, operation of typical interlace systems call for the two fields to be continuously refreshed alternately on the display screen. The results of this may be the appearance of the hands engaging in a jittery or shaking motion at the frequency of 30 times a second giving rise to highly undesirable images." This phenomenon is also illustrated in U.S. Patent No. 5,329,317 to Naimpally, et al.

These motion artifacts are particularly pronounced when the video in question has been magnified. For example, videos are often taken and magnified in endoscopic or laparoscopic surgery. Video images are taken by a high resolution camera coupled to an endoscope optic or laparoscope optic. Such images are magnified tremendously by the scope optic and, as a result, the captured images are extremely sensitive to motion. Thus, a small movement in an image field results in a much larger change in the viewed field, and such magnified motion appears to be more global than local. In addition, because of this magnification, such motion is exaggerated by the time separation between the two field components of the captured frame.

Motion effects in captured frames are created in at least two different ways, each of the at least two ways resulting in different types of motion artifact. One type of motion effect is generated by movement of the endoscope camera by the surgeon. Endoscope movement can result in a "uniform motion-related error" between the odd and even fields. This type of motion effect is known generally as "linear artifact." As

both fields are captured, endoscope movement causes a nearly identical image to be shifted in direct proportion to the velocity of endoscope movement, thus producing linear artifact. A second type of motion effect is created by movement within the image field of the camera. Such motion may be due to movement of surgical tools by the surgeon or by movement of the patient, such as with breathing. This type of motion effect is localized to the region of the image in which movement of the surgical tools or the patient tissue is being viewed, and is known as "regional artifact." The substantial magnification by the endoscope or laparoscope optic exacerbates the distorting effect of motion in a captured image caused by both linear artifacts and regional artifacts.

In sensitive applications, such as surgery, it is important to provide the most stable and artifact-free image possible. Efforts have been made in the past to correct for these motion artifacts. For example, video printers available from manufacturers such as Sony have "motion check" firmware. That firmware looks for motion artifacts developed within a particular image and alerts the user when it finds them. The user can then correct for the motion artifact. The correction usually involves dropping one of the two fields forming the image, which greatly reduces the vertical resolution of the video being displayed or printed. Another conventional technique for correcting for motion is to drop one of the two fields and replace the discarded field by repeating the remaining field. This results in an image that exaggerates only half the captured information, resulting in lower resolution. Also, some commercial software applications have motion correction features that can be performed by the user, although the

features are often difficult to implement by those not technically versed in its use. Adobe has such software.

Another approach to correcting for motion artifacts has been to compare the difference between pixels in two adjacent fields to a fixed threshold value. If the threshold is exceeded, then the value in one pixel is replaced. The replacement value may be determined by averaging the value of pixels in adjacent lines. If, however, the difference between pixels does not exceed the fixed threshold, no action is taken to change a pixel value. This process is repeated for each pixel in each line of each frame. An example of this approach as applied to motion artifacts within an endoscopic image field is described in U.S. Patent No. 5,877,819 to Branson.

Conventional methods for correcting for motion artifacts in highly magnified videos do not result in the highest quality picture. Moreover, they often require users with specialized skills, and lack flexibility. Thus, there is a need for a method and system for correcting for motion artifacts in highly magnified videos that yields a high resolution image. There is also a need for such a method and system that is dynamic with respect to pixel value thresholds so as to increase flexibility in further pixel value analysis and replacement. Moreover, there is a particular need for such methods and systems in sensitive applications, such as surgery.

SUMMARY OF THE INVENTION

The present invention provides methods and systems for improving video and images taken from endoscopic, or other medical and surgical procedures, sports still

frames, or other video in which the image recording device, subject, or both are moving independently. An example of an embodiment of the present invention described herein is a method for automatic correction of motion artifacts in an interlaced video image captured by an image recording camera. Such an embodiment may include
 5 capturing a complete frame of an interlaced video image, automatically correcting for camera motion, automatically correcting for subject motion, and displaying an image corrected for camera motion and subject motion.

In an embodiment, the method of the present invention involves first determining whether the image has been affected by camera motion. Since an entire image is
 10 affected by linear artifacts caused by camera motion, such camera motion artifacts must be removed from the image as a whole before the image can be examined for regional motion. Camera motion is identified and measured using auto-correlation of the odd and even raster fields. A two-dimensional vector is then calculated in order to provide optimal correction for camera motion, after which the even field may be
 15 repositioned into its correct location relative to the odd field (or vice versa) to correct for camera motion during the 1/60 second interval between capture of the two fields. This aspect of the invention corrects for situations such as "linear artifact" in which the entire camera itself moves relative to the viewed subject (or vice versa).

In another aspect of the invention, a method is provided for automatically
 20 identifying segments within the particular video image that contain subject, or regional, motion artifacts. Such a method of the present invention first determines where in an image subject motion has occurred. This may be done by convoluting the three red-

green-blue (RGB) color components of the odd field to compute or predict an even field. Differences between the measured even field and the predicted even field indicate regions of motion within the image field. When the difference between the predicted even field and actual even field is small, the difference can be attributed to subject
5 detail and is left uncorrected. When the difference between the predicted even field and the actual even field is large, subject, or regional, motion within the image field is identified.

The region of the image to which correction needs to be applied is then determined. One method of determining the region for correction is to form a subject motion map. In one implementation, a subject motion map may be generated by
10 squaring the differences between predicted even fields and actual even fields. Squaring the differences in pixel values between fields eliminates any positive or negative direction associated with those values, and makes the result less sensitive to small errors. The resulting subject motion map may then be compared to a threshold to
15 create a binary image delineating the region of the image in which regional motion artifacts have occurred.

Identified regional subject motion is then corrected. In one method, correction can be accomplished by computing the even field from the odd field in the region of the image where the subject motion map indicates low even field prediction accuracy. By
20 comparing identified regions of local artifact to the number of neighboring pixels above a set luminescence threshold, unwanted pixels may be eliminated one at a time.

Each region in the subject motion map where unwanted pixels have been eliminated may then be replaced but without the blurring caused by regional motion artifacts. In one method, auto-correction replacement for eliminated pixels can be accomplished by detecting adjacent pixels in two different directions. Such a two dimensional vector computed from non-blurred pixels at the boundaries of eliminated subject motion regions can be used to adjust the map by translating a sharp image therefrom into the region of subject motion.

A finished, corrected image can then be computed. In one method according to the present invention, a finished, corrected image is computed by using the adjusted map to indicate where to modify the original NTSC frame. Selective correction of the original frame may be accomplished by computing an even raster from the odd raster where the map indicates subject motion is greatest. After a finished image, automatically corrected for both camera motion and regional motion, is computed, the image is displayed for viewing.

One advantage of embodiments of the present invention is that motion artifacts, whether linear or regional, are automatically corrected. An advantage of regional field synthesis duplication as in the present invention is that there are no visible edge effects in the viewed image. In other words, regions of subject motion have smooth edges, rather than the jagged edges that often result from over-correcting.

Another advantage of automatically correcting only for regions that need correction as in the present invention is a high resolution image. Pure regional field duplication results in the loss of some image detail from discarding the even raster

pixels in the region of subject motion. Yet, in embodiments of the present invention, because pixels are discarded only in the regions needed, much fewer pixels are discarded than in systems that discard an entire field of pixels. Thus, the method and system of the present invention provide a flexible means for automatic correction of motion artifacts in highly magnified videos that result in the highest resolution-quality picture.

Embodiments of this invention automatically detect the type of artifact encountered in a captured frame on a pixel by pixel basis. Systems according to the present invention automatically identify whether the motion artifact is linear artifact caused by relative camera movement or regional artifact in the image field caused by subject (surgical instrument or patient) movement. By utilizing different parameters for detecting different types of motion, corrective action is taken only for that type of motion detected.

After automatically identifying the type of artifact, embodiments of this invention then automatically correct the frame image using the correct compensation for the particular artifacts encountered on each pixel of the frame. These benefits avoid any need by the surgeon to determine at the time of video capture whether motion correction is needed, and thus avoid prolonging surgical procedure time. In addition, an advantage of this system is that it allows the surgeon to capture still images without having to concentrate on assuring that the camera, the surgeon, surgical tools, and the patient are all still at the same instant each time an image is captured. Images can be

captured essentially at random without concern for motion artifacts caused by movement of any of these potential sources of artifacts.

Another advantage of the present invention is that embodiments can be used in the operating room, eliminating the time otherwise required to make post-procedure
5 image corrections. Non-automatic motion correction procedures typically involve the surgeon manually editing captured images by manipulating images with functions such as "motion up" and "motion down." The automatic correction for motion artifacts of this invention eliminates any need for manual editing.

Further, the minimal time required to apply the automatic motion correction of
10 this invention renders practical real-time image correction. This process can be applied, in real-time, using dedicated image processing equipment with the process implemented in dedicated circuitry or special purpose computational elements. Alternatively, this process can be applied by using a general purpose computer to compute corrections for motion artifacts using the process implemented in software.

15 Methods and systems for automatic motion correction according to the present invention may be used in a wide variety of applications in which video and images are taken from endoscopic or other surgery, sports still frames, or other video in which the camera and subject are moving independently. The present invention may be used with various video signal formats, including NTSC, PAL, and other sources of interlaced
20 video signals.

Therefore, the method and system of the present invention provide for automatic motion correction in video and images in which the camera, subject, or both move

independently that results in a high quality-resolution image. This is accomplished by maintaining primary information from both raster fields in an interlaced frame. The method and system of the present invention automatically and differentially detect linear motion artifacts caused by camera movement and regional motion artifacts caused by movement of surgical tools or the patient within the image field, both types of motion which may be occurring simultaneously. Using the method and system of the present invention, each type of motion artifact may be automatically and differentially corrected.

Those of ordinary skill in the art will appreciate the advantages and features of the present invention as described above and as is apparent from the detailed description below.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is a schematic diagram illustrating a system for practicing automatic motion correction of the present invention.

FIGURE 2 is a flow diagram illustrating operation of an embodiment implementing the automatic correction system of the present invention.

FIGURE 3 is a more detailed flow diagram illustrating options for applying an automatic correction system of the present invention in operation of the embodiment depicted in Figure 2.

FIGURE 4 is a flow diagram illustrating methods for automatic motion correction of the present invention as embodied in the system depicted in Figure 1.

FIGURE 5 is a more detailed flow diagram illustrating methods for automatic motion correction of the present invention as embodied in the system depicted in Figure 4, also showing steps to achieve automatic correction for camera motion and automatic correction for subject motion.

5 FIGURE 6 is a photograph depicting image distortion due to linear artifacts caused by camera motion.

FIGURE 7 is a photograph of the image of Figure 6 after automatic motion correction of camera motion linear artifacts.

FIGURE 8 is a photograph depicting image distortion due to camera motion and regional subject motion, as identified by a subject motion map.

FIGURE 9 is a photograph of the image of Figure 8 after automatic motion correction for camera motion and automatic correction for regional subject motion.

DETAILED DESCRIPTION

15 Methods and systems for automatic motion correction according to the present invention may be used in a variety of applications. Examples of applications include video and images taken from endoscopic, or other medical and surgical procedures, sports still frames, or other video in which the camera and subject are moving independently. The present invention may be used with various video signal formats, including NTSC, PAL, and other sources of interlaced video signals. In addition, the present invention may be used with a variety of hardware applications.

An example of an embodiment of the present invention is a method for automatic correction of motion artifacts in an interlaced video image captured by an image recording camera. Such an embodiment includes capturing a complete frame of an interlaced video image, the complete frame having a first raster field interlaced with a second raster field and both fields comprising pixels. Such an embodiment also includes automatically correcting for camera motion, automatically correcting for subject motion, and displaying an image corrected for camera motion and subject motion.

Methods of an embodiment of the present invention also utilize the steps of locating optimal correlation values between pixels in the first and second raster fields and creating a two-dimensional motion vector from such optimal correlation values. To achieve correction for camera motion, this embodiment then creates a synthetic first raster field by duplicating the second raster field in the captured complete frame in a corrected position according to the two-dimensional motion vector.

To accomplish correction for subject motion, methods of an embodiment of the present invention may also utilize the step of computing a subject motion map to identify regions of the captured frame where differences in pixel values between the first and second raster fields are relatively large. From a subject motion map, a binary subject motion location map may be created to delineate regions of the captured frame identifying pixels where the number of neighboring pixels exceeds a pre-determined threshold. In this embodiment, the pixels identified in regions of subject motion may then be eliminated. Pixels eliminated from regions of subject motion may then be replaced, but without blurring, by computing a two-dimensional vector from pixels at the

boundaries of eliminated regions. In this embodiment, a finished, corrected image may be created by using the adjusted binary subject motion location map to indicate regions on the captured frame where subject motion is greatest, and computing a corrected second raster field from a corrected first raster field in regions where subject motion is greatest. Methods of the embodiments described above may also utilize the step of displaying the image corrected for camera motion and subject motion on a video monitor.

In an application of the present invention in endoscopic surgery, one embodiment uses the "Dyonics® Vision 625 Digital Capture System" by Smith & Nephew, Inc. of Andover, Massachusetts. The "Dyonics® Vision 625 Digital Capture System" ("Dyonics® Vision 625") is designed to be used in the operating room to digitally capture intra-operative images. Images are stored in standard image file format on transportable media for long term archival, retrieval, or printing.

An embodiment of automatic motion correction for video images according to the methods and systems of the present invention is described using the "Dyonics® Vision 625," for example, in endoscopic surgery, in the steps below. FIGS. 1-4, in particular, depict operation of such an embodiment for practicing automatic motion correction as in the present invention.

1. Capture a complete frame (Frame 1).

The methods and system of automatic motion correction of the present invention operate on an interlaced video image comprising a complete frame having both even and odd fields captured. Referring to FIG. 5, for example, a complete NTSC frame 501

may be captured using a standard video capture device. In an embodiment using the "Dyonics® Vision 625" system 100, endoscope optic 101, as seen in FIG. 1, detects a video image in which both camera motion and regional subject motion are automatically detected and corrected. Referring to FIG. 1, an interlaced video signal is transmitted from endoscope optic 101 through camera control unit 102 to digital capture unit 103, where a full frame, interlaced video image is captured.

Digital capture unit 103 may have a keyboard 104 attached, which may be used to activate various modalities of the system, such as "freeze" and "capture," as seen at references 306 and 308, respectively, in FIG. 3. Live video images detected by endoscope optic 102 may be viewed (305a, 305b) as displayed on video monitor 105. In operation, the surgeon may activate a "freeze" signal 110 at the camera control unit 102 to freeze (306) a live video image (305a) and display the frozen image (307) on video monitor 105. As the surgeon views a live image (305a), the image may be captured by activating a "capture" signal at the camera control unit 102, the image being saved (309) to the internal temporary storage 112 of digital capture unit 103. By activating "bypass" mode 330, the digital capture unit 103 is deactivated, or suspended from application to the live video images being viewed, and live images are displayed directly on video monitor 105.

Automatic correction for camera motion 405 and automatic correction for regional subject motion 406, depicted in greater detail in FIG. 5, are operationalized by Automatic Motion Correction (AMC) Filter 109 of the embodiment shown in FIG. 1.

Automatic correction for both camera motion and for regional subject motion are described below.

2. Perform auto-correlation on a first raster field with respect to a second raster field (Frame 1).

5 The first automatic motion correction performed by the methods and system of the present invention is for camera motion. FIG. 6 illustrates image distortion due to linear artifacts caused by camera motion. Correction for camera motion is accomplished by performing auto-correlation on a first raster field, either the even field or the odd field, with respect to the second raster field, the complementary field to the first field, in the captured frame. The auto-correlation function results in a "motion vector." Using the "Dyonics® Vision 625" system, the auto-correlation motion vector is computed by performing the function "MeasureMotionVector" 502 in FIG. 5. "MeasureMotionVector" 502 outputs two values through the argument list SkewX and SkewY. To save time, the algorithm uses an iterative search procedure, minimizing a 3x3 convolution. The best-fit location of the 3x3 result matrix is used to search at a new position until an optimal match is found, or the first raster is determined to be greater than 15 pixels offset from the second raster.

"MeasureMotionVector" 502 calls one subroutine, "Convolution" 503. The purpose of "Convolution" is to save results from successive computations to eliminate duplication of computations already performed in previous searches for the optimum auto-correlation point. Generally, "ConvolveAt" is a mathematical technique that

compares two images by generating a single number for each of various X and Y points indicative of the quality of the fit of each test pixel in one raster relative to the reference pixel in the other raster.

In the present invention, "ConvolveAt" convolves the difference of all pixels between the odd and even rasters for a specified offset. When the specified offset is zero for both X and Y, the same pixels in both rasters are compared to each other. When the offset is one X and zero Y, the even raster pixel to the right of the same pixel in the odd raster is compared. The convolution in "ConvolveAt" measures the sum of the square of the differences between the rasters at the specified offset. The optimum auto-correlation between the odd and even rasters is the value of X and Y which produces a minimum difference between the two rasters. By determining the optimum auto-correlation points for shifting the test image relative to a reference image, convolution creates as sharp a full-frame image as possible. The convolution step corrects for linear artifacts caused by camera motion, as shown by reference 405 in FIG. 4. Yet, image differences between the two raster fields due to regional artifacts may still be present at this point.

The following algorithm applies to auto-correlation on a first raster field with respect to a second raster field. However, as will be apparent to those of ordinary skill in the art, the automatic motion correction method and system of the present invention are also readily utilized in applications that do not employ this convolution technique.

```
void MeasureMotionVector(unsigned char *Fields, long *SkewX, long *SkewY)
{
    unsigned long Data[33][33] = {0};
    int x, y, NextX, NextY;
```

```

unsigned long Center, t, l, b, r;
double Szy2, Szy, Sz, A, B, DeltaY;

```

```

NextX = 0;    // kick start value!
5  x = 1;    // kick start value!
NextY = y = 0;

```

```

while((x != NextX) || (y != NextY))

```

```

{
10   x = NextX;
    y = NextY;

```

```

    Center = Convolution(Data, Fields, x, y);
    t = Convolution(Data, Fields, x, y-1);
15   l = Convolution(Data, Fields, x-1, y);
    b = Convolution(Data, Fields, x, y+1);
    r = Convolution(Data, Fields, x+1, y);

```

```

    if(Center > t)

```

```

    {
20       NextX = x;
        NextY = y-1;
        Center = t;
    }

```

```

    if(Center > l)

```

```

    {
25       NextX = x-1;
        NextY = y;
        Center = l;
    }

```

```

    if(Center > b)

```

```

    {
30       NextX = x;
        NextY = y+1;
        Center = b;
    }

```

```

    if(Center > r)

```

```

    {
40       NextX = x+1;
        NextY = y;
    }

```

```

    Center = Convolution(Data, Fields, x, y);

```

```

    if(x == 15) break;

```

```

    if(x == -15) break;

```

```

45   if(y == 15) break;

```

```

    if(y == -15) break;

```

```

    if(!Center && !t && !b)

```

```

    {
50       *SkewX = 0;
        *SkewY = 0;
        return;
    }

```

```

}

```

```
// using least squares, interpolate Y from t(op), Center, and b(ottom)
// note: Center is at zero pixels offset, t is -2 and bottom is 2
```

```
// using notation: z is a function of y... and the 2nd order poly coefficients a, b, c
```

```
5 // where  $z = ay^2 + by + c$ 
//  $f(a, b, c) = \text{Sigma}(((ay^2 + by + c) - z)^2)$ ;
// minimize  $f(a, b, c)$ 
// take partial derivatives:
//  $df/da = 2 \text{Sigma } y((ay + by + c) - z)$ 
10 // 1a)  $df/da = 2 \text{Sigma } (ay^4 + by^3 + cy^2 - zy^2)$ 
```

```
//  $df/db = 2 \text{Sigma } y((ay + by + c) - z)$ 
// 2a)  $df/db = 2 \text{Sigma } (ay^3 + by^2 + cy - zy)$ 
```

```
15 // 3a)  $df/dc = 2 \text{Sigma } (ay^2 + by + c - z)$ 
```

```
// setting the partials to zero to find the minima...
```

```
// 1b)  $\text{Sigma } zy^2 = a \text{Sigma } y^4 + b \text{Sigma } y^3 + c \text{Sigma } y^2$ 
```

```
// 2b)  $\text{Sigma } zy = a \text{Sigma } y^3 + b \text{Sigma } y^2 + c \text{Sigma } y$ 
```

```
20 // 3b)  $\text{Sigma } zz = a \text{Sigma } y^2 + b \text{Sigma } y^1 + nc$ , (note  $n=3$ )
```

```
// compute the matrix coefficients:
```

```
Szy2 = t * 4.0 + Center * 0.0 + b * 4.0;
```

```
Szy = t * -2.0 + Center * 0.0 + b * 2.0;
```

```
25 Sz = (double)t + (double)Center + (double)b;
```

```
//  $Sy4 = 32 = 16.0 + 0.0 + 16.0$ ;
```

```
//  $Sy3 = 0 = -8.0 + 0.0 + 8.0$ ;
```

```
//  $Sy2 = 8 = 4.0 + 0.0 + 4.0$ ;
```

```
//  $Sy = 0 = -2.0 + 0.0 + 2.0$ ;
```

```
30 // the matrix is:
```

```
// 1c)  $Szy2 = a*32.0 + b*0.0 + c*8.0$ 
```

```
// 2c)  $Szy = a*0.0 + b*8.0 + c*0.0$ 
```

```
35 // 3c)  $Sz = a*8.0 + b*0.0 + c*3.0$ 
```

```
// To find the y location of the
```

```
// parabolic vertex: minimize  $dz/dy = 2a\Delta Y + b$ 
```

```
// so  $\Delta Y = -b/2a$ , therefore we only need to solve the matrix for a and b.
```

```
// from eqn 2c:
```

```
40 B = Szy / 8.0;
```

```
// now with two eqns in 2 unknowns:
```

```
// 1d)  $Szy2 = a*32.0 + c*8.0$ 
```

```
// 3d)  $Sz = a*8.0 + c*3.0$ 
```

```
45 // normalize c
```

```
// 1e)  $Szy2/8.0 = a*32.0/8.0 + c$ 
```

```
// 3e)  $Sz/3.0 = a*8.0/3.0 + c$ 
```

```
// rearrange
```

```
// 1f)  $Szy2/8.0 - a*32.0/8.0 = c$ 
```

```
50 // 3f)  $Sz/3.0 - a*8.0/3.0 = c$ 
```

```
// solve for a:
```

```
//  $Szy2/8.0 - a*32.0/8.0 = Sz/3.0 - a*8.0/3.0$ 
```

```
//  $Szy2/8.0 - Sz/3.0 - a*32.0/8.0 = -a*8.0/3.0$ 
```

```
//  $Szy2/8.0 - Sz/3.0 = a*32.0/8.0 - a*8.0/3.0$ 
```

```
// Szy2/8.0 - Sz/3.0 = a*(32.0/8.0 - 8.0/3.0)
// (Szy2/8.0 - Sz/3.0)/(32.0/8.0 - 8.0/3.0) = a
A = (Szy2/8.0 - Sz/3.0)/(32.0/8.0 - 8.0/3.0);
```

```
5 // so finally:
DeltaY = -B/(2.0 * A);

// report the results to the calling function
*SkewX = -x;
10 *SkewY = -y*2 - DeltaY*2; // round instead if truncate
}
```

```
unsigned long Convolution(unsigned long Data[33][33],
    unsigned char *Fields, long x, long y)
```

```
15 {
// already computed?
if(Data[y+16][x+16])
    return Data[y+16][x+16];
Data[y+16][x+16] = ConvolveAt(Fields+640*3*240, Fields, x, y);
20 return Data[y+16][x+16];
}
```

```
unsigned long ConvolveAt(unsigned char *Odd, unsigned char *Even, long x, long y)
```

```
{
25 long i, j, Sigma = 0;
```

```
x += 16;
y += 16;
```

```
30 Odd += 16 * 640 * 3 + 16 * 3;
Even += y * 640 * 3 + x * 3;
```

```
for(i=0; i<240-33; i++)
```

```
35 {
    j = (640-33)*3;
    while(j-->0)
    {
        Sigma += (*Odd - *Even) * (*Odd - *Even);
        Odd++;
        Even++;
    }
    Odd += 33*3;
    Even += 33*3;
}
```

```
45 return Sigma;
}
```

3. Create a synthetic first raster field by duplicating the second raster field and then
50 placing the synthetic first raster field into its corrected position (Frame 2).

To complete automatic motion correction for camera motion, an intermediate frame, substantially transparent to the human eye, is created. Figure 7 illustrates the image of Figure 6 after automatic correction of linear artifacts due to camera motion.

5 This intermediate frame comprises a raster with a synthetic first raster field generated by duplicating the second raster field and then placing the synthetic first raster field into its corrected position. In an embodiment of the methods and system of the present invention, using the "Dyonics® Vision 625" system, such a synthetic field is generated and placed into its corrected position by automatically running the

10 "ShiftEvenAndJoinOddField" function 504, as depicted in FIG. 5. The function "ShiftEvenAndJoinOddField" 504 replaces the even field in the NTSC frame at the offset determined by the function "MeasureMotionVector" 502. The following algorithm applies to a process of creating a raster with a synthetic field.

```

15 void ShiftEvenAndJoinOddField(unsigned char *Fields, long x, long y, unsigned char *RawImage)
{
    unsigned char *Odd, *Even, *WorkingImage;
    long i, n, Lines, Skip, Shift, Width;

20 // blindly fill both rasters with odd
    WorkingImage = RawImage;
    Even = Odd = Fields + 640*3 * 240;
    for(i=0; i<240; i++)
    {
25         n = 640*3;
        while(n--)
            *WorkingImage++ = *Even++;
        n = 640*3;
        while(n--)
30             *WorkingImage++ = *Odd++;
    }

    // compute refill values...
    if(y>0)
35     {
        Lines = 240-y/2;
    }
}

```

```

Even = Fields;
WorkingImage = RawImage + 2*640*3*(y/2);
}
else
5   {
    Lines = 240+(y-1)/2;
    Even = Fields + 640*3*(-(y-1)/2);
    WorkingImage = RawImage;
    }
10  if(y%2)
    {
        WorkingImage += 640*3;      // exchange
    }

15  if(x>0)
    {
        Skip = (640+x)*3;
        Shift = x*3;
        Width = (640-x)*3;
        WorkingImage += Shift;
20  }
    else
    {
        Skip = (640-x)*3;
        Shift = -x*3;
        Width = (640+x)*3;
        Even += Shift;
25  }

30  // now join the rasters...
    for(i=0; i<Lines; i++)
    {
        n = Width;
        while(n--)
35  *WorkingImage++ = *Even++;
        WorkingImage += Skip;
        Even += Shift;
    }
}

40

```

4. Compute a subject motion map (Frame 3).

Figure 8 illustrates image distortion due to camera motion and regional subject motion. After camera motion is automatically corrected according to the methods and system of the present invention, regional subject motion can then be automatically

detected and corrected, as shown at reference 406 in FIG. 4. Referring to FIG. 5, regional subject motion can be automatically detected by computing a subject motion map, which comprises another transparent, or non-viewable, intermediate frame, Frame 3. Subject motion can be found anywhere in an image after camera motion effects are removed. The function "LocateArtifacts" 505, using the "Dyonics® Vision 625" system, for example, looks for the results of subject motion in the difference between even and odd rasters. To compute the difference between the "expected" even pixel value and the actual even pixel value, the "LocateArtifacts" 505 function uses reference images of the convolution kernel:

0 1 0
0 -20
0 1 0.

However, any equivalent convolution kernel could be used, such as:

1 0 1
0 -40
1 0 1

or

1 1 1
0 -60
1 1 1.

The kernel is chosen for speed of execution. The three active terms are adequate because the even raster of the image already has the greatest possible correlation to the odd raster because of the processes described above to create a raster with a synthetic field 504 and placing the synthetic field into its corrected position (frame 2).

"LocateArtifacts" 505 convolves the NTSC frame "ColorRaster" to produce a half height grayscale image in "Diff". The following algorithm applies to the process for computing a subject motion map.

```

5  int absdif(int x, int y)
    {
      int z;

      z = x - y;
10  return z < 0 ? -z : z;
    }

// color raster is 640x480x3 Diff is 640x240x1
void LocateArtifacts(unsigned char *ColorRaster, unsigned char *Diff)
15 {
  unsigned char *Top, *Center, *Bottom;
  int y, x, c;
  unsigned int Avg, DTB;
  int Delta;

20  for(y=0; y<480; y += 2)
    {
      Top = ColorRaster + y * 640*3;
      Center = Top + 640*3;
      Bottom = Center + 640*3;

25      x=640;
      while(x--)
        {
          Delta = 0;

          c=3;
          while(c--)
          {
35              DTB = absdif(*Top, *Bottom);
              Avg = *Top++;
              Avg += *Bottom++;
              Avg /= 2;
              Delta += absdif(Avg, *Center++);
              Delta -= DTB / 2;
40          }
          Delta -= 15;
          if(Delta < 0) Delta = 0;
          Delta *= Delta;
45          *Diff++ = Delta < 256 ? Delta : 255;
        }
      }
}

```


5. Establish a threshold difference and eliminate unwanted pixels in Frame 3 to create a binary subject motion location map (Frame 4).

After a subject motion map is computed, a binary subject motion location map is generated to correct for regional motion artifacts by eliminating undesired pixels and yet preserve actual detail in the affected region. In the methods and system of the present invention, using the "Dyonics® Vision 625" system embodiment, for example, a binary subject motion location map is generated by the function "Decimate" 506 (FIG. 5). The function "Decimate" 506 compares each pixel in the grayscale image "map" to a threshold difference determined by experimenting with a large number of images. In practical application using the convolution kernel in "LocateArtifacts" 505, a threshold difference anywhere from about 20 to about 150 IRE brightness units works well. A preferred final average threshold difference value is 80 IRE units.

"Decimate" 506 performs binary decimation on the subject motion map to eliminate hairline signals resulting from high image detail. Decimation eliminates unwanted pixels one at a time, and is performed by first counting the number of neighboring pixels exceeding the threshold. If more than three neighboring pixels in the subject motion map are above the threshold, the pixel is not decimated. If three or less neighboring pixels in the subject motion map are above the threshold, the pixel is eliminated. As such, local detail is preserved, while motion artifacts caused by surgical tool or patient movement is corrected. The following algorithm applies to the process of decimation to create a binary subject motion location map.

```

void Decimate(unsigned char *Map)
{
int Total, Count = 640 * (240 - 2) - 2;
unsigned char *Previous, *Current, *Next;
5  unsigned char Threshold = 80;

Previous = Map;
Current = Previous + 640;
Next = Current + 640;
10 while(Count--)
{
Total = 0;
Total += Previous[-1] < Threshold ? 0 : 1;
Total += Previous[0] < Threshold ? 0 : 1;
15 Total += Previous[1] < Threshold ? 0 : 1;
Total += Current[-1] < Threshold ? 0 : 1;
Total += Current[1] < Threshold ? 0 : 1;
Total += Next[-1] < Threshold ? 0 : 1;
Total += Next[0] < Threshold ? 0 : 1;
20 Total += Next[1] < Threshold ? 0 : 1;

*Current = Total < 3 ? 0 : *Current; // leave alone
Previous++;
Current++;
25 Next++;
}
}

```

6. Adjust the binary subject motion location map (Frame 5).

After establishing a threshold and decimating unwanted pixels to create a binary subject motion location map in intermediate frame 4 in the previous step, the binary map image is adjusted in a subsequent intermediate frame to replace decimated areas but without the blurring caused by motion artifacts. In the methods and system of the present invention, using the embodiment of the "Dyonics® Vision 625" system, for example, the map is "dilated" to identify pixels near the boundaries of regions eliminated due to subject motion artifacts. Pixels near the boundaries are most likely to be affected by similar subject motion. The "Dilate" routine 507 (FIG. 5) replaces decimated areas but without the blurring caused by regional motion artifacts. The

"Dialate" routine 507 works in similar fashion to the "Decimate" routine 506, except the effect is to "spread" the map out one pixel each time the routine is called. A two-dimensional vector is computed by detecting pixels in two directions, one pixel at a time, adjacent to pixels identified at the boundaries of eliminated regions. Pixels eliminated within regions of subject motion are then replaced with the two-dimensional vector to create a corrected image having smooth edges. The "Dialate" routine 507 is automatically run twice to enhance image sharpness. The following algorithm applies to the process for adjusting the map at pixels near the threshold.

```

10 void Dialate(unsigned char *Map)
   {
   int Total, Count = 640 * (240 - 2) - 2;
   unsigned char *Previous, *Current, *Next;
   unsigned char Threshold = 80;

   Previous = Map;
   Current = Previous + 640;
   Next = Current + 640;
   while(Count--)
   {
       Total = 0;
       Total += Previous[-1] < Threshold ? 0 : 1;
       Total += Previous[0] < Threshold ? 0 : 1;
       Total += Previous[1] < Threshold ? 0 : 1;
       Total += Current[-1] < Threshold ? 0 : 1;
       Total += Current[1] < Threshold ? 0 : 1;
       Total += Next[-1] < Threshold ? 0 : 1;
       Total += Next[0] < Threshold ? 0 : 1;
       Total += Next[1] < Threshold ? 0 : 1;

       *Current = Total > 4 ? 255 : *Current; // leave alone
       Previous++;
       Current++;
       Next++;
   }
   }

```

7. Compute a finished, corrected image (Frame 6).

After the binary subject motion location map image is adjusted by replacing decimated areas but without blurring caused by regional motion artifacts, a finished, corrected image is computed. A finished, corrected image is computed by using the adjusted map (frame 5) to indicate where to modify the original NTSC frame ColorRaster. In the methods and system of the present invention, using the embodiment of the "Dyonics® Vision 625" system, for example, the original frame ColorRaster is modified by running the "InterpolateMap" routine 508, as shown in FIG. 5. The "InterpolateMap" routine 508 selectively corrects the ColorRaster by computing an even raster from the odd raster where the map indicates subject motion is greatest. That is, local regions of the image field having large amounts of subject motion are selected out for correction. As a result, a sharp finished image is interpolated. FIG. 9 illustrates the image of Figure 8 after automatic motion correction. The following algorithm applies to a process of selectively correcting the original frame ColorRaster to produce a finished image, corrected for both camera motion and regional subject motion.

```

// color raster is 640x480x3 Map is 640x240x1
void InterpolateMap(unsigned char *ColorRaster, unsigned char *Map)
{
20  unsigned char *Top, *Center, *Bottom;
    int y, x, c;
    unsigned int Avg, DTB;
    int Delta;

25  Top = ColorRaster;
    Center = Top + 640*3;
    Bottom = Center + 640*3;

    y = 240;
30  while(y--)
    {
        x = 640;

```

```

while(x--)
{
    if(*Map++ > 128)
    {
        Avg = *Top++;
        Avg += *Bottom++;
        *Center++ = Avg >> 1;
        Avg = *Top++;
        Avg += *Bottom++;
        *Center++ = Avg >> 1;
        Avg = *Top++;
        Avg += *Bottom++;
        *Center++ = Avg >> 1;
    }
    else
    {
        Top += 3;
        Center += 3;
        Bottom += 3;
    }
}
Top += 640 * 3;
Center += 640 * 3;
Bottom += 640 * 3;
}

```

As a result of operation of steps 505-508 in FIG. 5 (frames 3-6), the original captured frame (frame 1) is automatically corrected for regional subject motion, as indicated by step 406 in FIG. 4.

8. Display the finished, corrected image (Frame 7).

After a finished image 509 (FIG. 5), automatically corrected for both camera motion and regional subject motion, is computed, the image is displayed for real-time viewing, by the surgeon for example. Images automatically corrected according to the methods and system of the present invention, using the embodiment of the "Dyonics® Vision 625" system, for example, may also be stored in standard image file format on transportable media for long term archival, retrieval, or printing. Referring to FIGS. 1

5

0